

Nama : Lius Harsen

Kelas : 2IA11

NPM : 50422810

## Rangkuman Pertemuan 1-6

### Pengenalan pemrograman berbasis objek

- Pemrograman berorientasi objek adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat.
- Pemrograman berorientasi objek ditemukan pada Tahun 1960, dimana berawal dari suatu pembuatan program yang terstruktur . Metode ini dikembangkan dari bahasa C dan Pascal. Dengan program yang terstruktur inilah untuk pertama kalinya kita mampu menulis program yang begitu sulit dengan lebih mudah
- Pemrograman berorientasi objek bekerja dengan baik ketika dibarengi dengan ObjekOriented Analysis And Design Process .
- Struktur program ringkas, cukup dengan membuat Objek dan class lalu bekerja berdasarkan object dan class tersebut.
- Kode program sangat re-usable.object dan class dapat digunakan berkali-kali,sehingga dapat menghemat space memori.
- Class adalah cetak biru atau blueprint dari object. Class digunakan hanya untuk membuat kerangka dasar.
- Sebagai analogi, class bisa diibaratkan dengan kendaraan. Kita tahu bahwa kendaraan memiliki ciri-ciri seperti merk, memiliki mesin, memiliki ban, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah kendaraan. Selain memiliki ciri-ciri, sebuah kendaraan juga bisa dikenakan tindakan, seperti: menghidupkan mesin atau mematikan mesin.
- Objek mirip dengan suatu rekaman dalam suatu sistem berkas .
- Objek adalah entitas yang memiliki atribut, karakter dan kadang kala disertai kondisi. Objek mempresentasikan sesuai kenyataan seperti siswa, mempresentasikan dalam bentuk konsep seperti merek dagang, juga bisa menyatakan visualisasi seperti bentuk huruf .

### Inheritance

- Pewarisan merupakan sebuah bentuk «penggunaan kembali» ; dimana class baru dibuat dari class yang pernah ada yang ditambah fasilitasnya.
- Setiap class turunan dapat menjadi class pokok untuk class turunan yang akan datang.
- Dalam pewarisan, constructor tidak diwariskan pada class turunannya, kecuali jika digunakan perintah super.
- Pewarisan tunggal merupakan pewarisan dari satu class pokok .
- Pewarisan ganda merupakan pewarisan dari dua atau lebih class pokok.
- Java tidak mendukung multiple inheritance..
- Class yang mewariskan disebut dengan superclass / parent class / base class, sedangkan class yang mewarisi disebut dengan subclass / child class / derived class.

### Encapsulation

- Enkapsulasi merupakan cara untuk melindungi property /method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program.

Accessors : - Cara untuk melindungi data yaitu dengan menggunakan access modifiers .

### **Polymorphism**

- Overloading adalah diperbolehkannya dalam sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda.-
- Overriding method adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk method superclass-nya.

### **Inheritance**

- Untuk setiap burung, ada satu set properti yang telah ditetapkan yang umum untuk semua burung dan ada satu set properti yang khusus untuk burung tertentu. Oleh karena itu, secara intuitif, kita dapat mengatakan bahwa semua burung mewarisi ciri-ciri umum seperti sayap, kaki, mata, dll. Oleh karena itu, dalam cara berorientasi objek untuk merepresentasikan burung, pertama-tama kita mendeklarasikan kelas «Bird» dengan seperangkat properti yang umum untuk semua burung. Dengan melakukan ini, kita dapat menghindari menyatakan sifat-sifat umum ini di setiap burung yang kita buat. Sebagai gantinya, kita cukup mewariskan kelas «Bird» di semua burung yang kita buat.

### **Encapsulation**

- Sekarang, kita telah mendefinisikan properti dari kelas «Bird» dan atribut yang dimiliki burung seperti warna, sayap, kaki dapat diinisialisasi dengan membuat objek kelas burung. Namun, jika kita hanya dapat mengubah properti dari kelas «Bird» hanya dengan referensi dari objek, maka atribut kehilangan informasi yang awalnya diinisialisasi.
- Misalnya, katakanlah kita awalnya membuat merpati dengan warna abu-abu dengan membuat konstruktor, setiap pengguna dengan instance objek merpati dapat mengubah warna ini menjadi merah atau hitam hanya dengan merujuk atribut dengan kata kunci «this». Untuk menghindari hal ini, kita menyertakan properti ke dalam sebuah metode. Metode ini disebut getter dan setter atribut. Identya adalah untuk hanya menyertakan inisialisasi dan pengambilan atribut dalam suatu metode alih-alih langsung merujuk atribut secara langsung. Ini juga memberikan keuntungan karena setter memberi kita kendali penuh dalam menetapkan nilai ke atribut dan membantu kita membatasi perubahan yang tidak perlu.

### **Polymorphism**

- Sekarang, kita ingin membuat merpati dapat terbang dengan menggunakan method «fly» namun kita ingin mendefinisikan tujuan terbang merpati tersebut dan kita ingin mendefinisikan makanan yang dimakan oleh merpati tersebut. Kita dapat melakukannya dengan menggunakan konsep overload method.

### **Testing**

- Setelah kita selesai membuat kelas «Bird» dan kelas «Pigeon», kita perlu membuat sebuah kelas untuk menjalankan atau memanggil kelas-kelas tersebut beserta method-methodnya.